

A Mobile Application for Voice Enabled Virtual Bot

Jimcymol James¹, Shabih Fatima², Sanjeevani Avasthi³, Chirag Nalawade⁴, Jenny James⁵,
Mirza Mohammad⁶, Suhana Ayisha⁷, Gautam Ravi⁸, Priya Shrotri⁹

Manipal Academy of Higher Education, Dubai, UAE.

Abstract

This technical paper presents the design and development of an Artificial Intelligence Voice Assistant for a University campus that can interact with the user and answer questions related to its domain of knowledge. It uses Artificial Intelligence to enhance user satisfaction by improving one to one communication. It will take the user's voice as an input, analyze the speech, and respond with appropriate answers to the user's questions by using technologies like Speech Recognition, Natural Language Processing and Text Analysis, Information Retrieval, Knowledge Representation, Speech Synthesis and Machine Learning.

Keywords : Virtual Assistant; Chatbot; Information Retrieval, Android Application, Artificial Intelligence; Voice Recognition and Generation; Natural Language Processing; Machine Learning; Enquiry bot; Dialogflow

INTRODUCTION

It is often quite hectic, time-consuming and inconvenient to acquire the relevant information about any organization or institution. Traditionally, queries such as these are handled through human to human interactions. Consequently, as the number of inquiries increases, the waiting time to answer these queries increases, which results in reduced customer satisfaction. With the recent developments in AI-based applications and Natural Language Processing, this process can be made much faster and simpler by creating an AI Voice Assistant to handle these tasks.

The goal of this paper is to provide a quick and simple solution to this problem. The Virtual Assistant maintains a natural conversation with the user. It can answer queries related to the infrastructure, courses, accounts, faculty, and events of our institute.

The Virtual Assistant receives a question from the user, understands the matter, and provides appropriate answers. It does this by converting an English sentence into a machine-friendly query, extracting the relevant keywords, then going through the necessary information and finally returning the answer in a natural language sentence. In other words, it answers the questions like a human does.

The Virtual Assistant provides a convenient solution to users that are new to the institution or need assistance in general. It also augments the productivity of the employees by automating the processes that do not necessarily require humanitarian aid such as answering Frequently Asked Questions.

By using the Virtual Assistant, everything a user needs to know about the institution is just one tap away.

PROJECT OVERVIEW

Requirement Gathering and Analysis

- *Requirement Gathering*

The requirements for the virtual assistant project were gathered by surveying the students at various universities, one on one interviews with the people at the help desk and people waiting in queue to get their queries answered at the university. We collected a database of information based on our survey. This database included the general details about the university - information about various courses, details of the faculty, aspects of the infrastructure and information regarding various events and clubs present in the university. [1]

- *Analysis*

To get an idea of the working of our application we investigated similar claims in use such as Siri[2] and Google Assistant[3]. The Analysis phase started off with our team interacting with the students and faculties and taking their inputs about our application and the features needed. Then further, we decomposed similar information into smaller parts and created a relationship between them. This phase also included the documentation of the project lifecycle and setting up the deadline for various milestones.

Design

This section discusses the architecture of the bot as given in Figure 1.

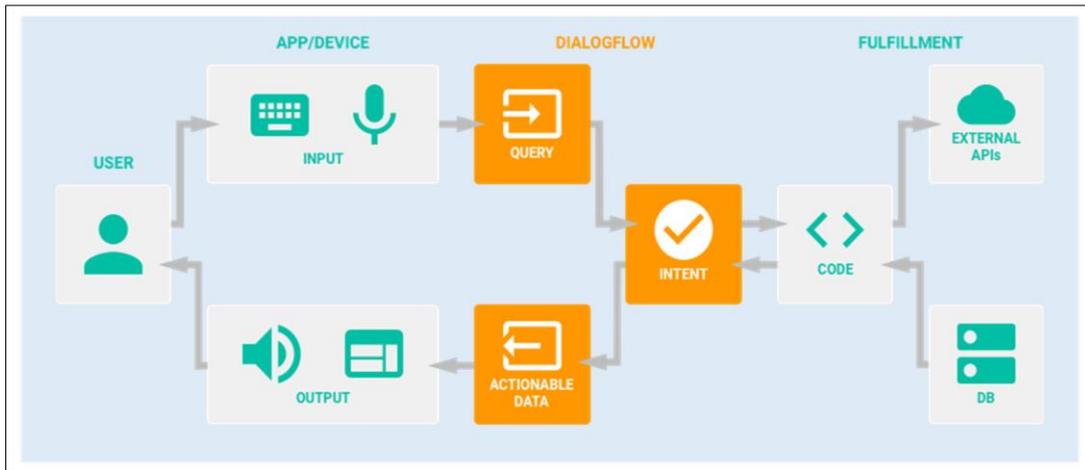


Figure 1. Architecture of the Voice Assistant [4]

Front-end

The front end of the application is a lightweight, simple and straightforward mobile application with an interface that is designed for the ease of use by all age groups and requires minimal technical experience.

The application interface consists of a single button for the bot to record the user input in the form of speech which is then converted into text. It also comprises two text views - one displays the user's query and the other displays the reply by the bot.

Back-end

The backend of the bot can be divided into three modules:

- *App module*

This module is responsible for taking the user input, converting it from speech to text and sending the query to Dialogflow[4]. The response from the bot is in the form of text which is then converted into speech.

- *Dialogflow module*

This module transforms the input into a query and matches it with the most appropriate intent. Intents are the components of the bot that process a user's request. Each intent has one type of question that could be asked by the user and the answers to it. For example, all the questions about faculty contact numbers will be mapped to one intent, and the intent will process the answers for the same. In other words, an intent represents a mapping between what a user says and what action should be taken by the bot.

An Intent has the following parts:

- *Training Phrases*: Training phrases contains various alternative ways a user may ask the question handled by the particular intent.
- *Action*: Action is used to identify the entities in the training phrases, mark what all entities are required to

answer the question and even define prompts to the user in case the user's input is not clear.

- *Response*: The response part contains the answers to the question. The solutions can be static or fetched from a web service by using a webhook.
- *Contexts*: Contexts represent the current context of a user's request. Contexts are essential in maintaining the flow of conversation in a dialog system.

The bot uses a webhook for fulfillment and retrieves the data for the response from an external database. It then returns the answer to the app module.

- *Fulfillment module*

This module consists of storing the information about various entities related to the university that the user could ask the bot. Entities are powerful tools used for extracting parameter values from natural language inputs. For example, in our bot, we have created entities such as "faculty_name" which stores all the names of the faculties and the different synonyms for each name. Hence, when the user says the name of a faculty, it will be recognized as one of the values in the "faculty_name" entity. This entity name will then be taken as a parameter value which will be sent to the web service to retrieve the appropriate answer to the user's question.

The data is stored in the form of tables that can be quickly updated and managed by the admin. This module also contains the code for the webhook that is responsible for querying the database based on the user's question.

Development

The development of the project can be divided into the following phases -

- Construction of the database
- Development of the bot on Dialogflow
- Development of the webhook
- Development of the Android app

Construction of the database

The database for the application is created using MySQL. It can be managed by using phpMyAdmin. The database tables were created and populated by using SQL commands.

Development of the bot on Dialogflow

- **Creating intents**

Intents can be uploaded to the Dialogflow platform in JSON format having all the required fields, or they can be created using the GUI of the platform. [5]

- **Creating entities**

An entity can be created by uploading entity file in CSV or JSON format. It can also be formed by using the platform. [6]

- **Training the bot**

The bot can be trained by uploading user inputs to the Training tool in a .txt file or a .zip archive with multiple .txt files. The bot can also be continuously trained while it is being used by the clients. The developer can review the responses of the bot at any time and approve them. In case the bot gives a wrong answer for a query, the developer can review and correct the intent mapping and hence train the bot.

- **Development of the webhook**

The webhook is a PHP script that accepts POST requests. When the user asks a question, the Dialogflow agent sends a POST fulfillment request to the webhook that contains the query and the parameters in a JSON format. The webhook accepts the POST request and parses the JSON object to extract the relevant information from the query. Based on the keywords, it queries the database and retrieves the answers that are sent to the Dialogflow agent as a JSON fulfillment response.

- **Development of the Android app**

The Android application was built on Android Studio v 3.1.

The details of the application are as follows: -

- Requires permission to access the internet
- Requires permission to record audio
- Minimum Requirement - API level 22: Android 5.1 Lollipop

Testing

Functional Testing

Each individual member of the team has inspected the bot with their own set of questions to verify that the desired output for each question is obtained.

Samples: -

- To find the identity of the academic president:

Input:

Who is the academic president?

Output:

Dr Raj is the Academic President of SOT Campus.

- To find the application fees for a course:

Input:

How much fees will it take to apply?

Output:

The application fees for any course is 100 Dirhams.

- To find the location of the Cisco Lab:

Input:

Where is the Cisco lab?

Output:

It is on the second floor, A block.

Stress Testing

The bot has been tested using a minimal number of keywords as input to determine if it can recognize which output is the most appropriate even under circumstances with similar questions that require different answers.

Samples: -

- To get the tuition fees for the Bachelor of Architecture course the following inputs are tested:

Inputs:

How much for Architecture

Architecture fees

Cost of architecture

Bachelor of Architecture fees

Bachelor Architecture

Output:

The tuition fees for Bachelor of Architecture is AED 42,000

Performance Testing

The expected output for each question is given within 5 seconds after completing the question.

Usability Testing

\The application was well received by outsiders as it is easy to use and does not require any technical knowledge.

Acceptance Testing

This application has been tested by several users and has been judged in competitions without any known faults found in its functioning.

Table 1. Sample Test Cases

Query	Response
“How are you”	“I’m doing well. Thank you. And you?”
“Where is the Account’s department”	“Go to A block, first floor.”
“What are the fees of engineering”	“The tuition fees is AED 42,000 per year.”
“Where is the washroom in Block A”	“The washroom is on the right side of the lift.”
“Where is SOT located”	“It is located in Academic City, Dubai.”
“Who are you”	“Your virtual assistant, MaheBot.”
“When is Technovanza”	“Technovanza is to be held on 11 th and 12 th of April 2018”
“How many blocks does SOT dubai have”	“SOT has three blocks – Block A, Block B and Block C”
“Where is Professor John’s cabin”	“Professor John’s cabin is in A block, 3 rd floor, opposite class 405.”

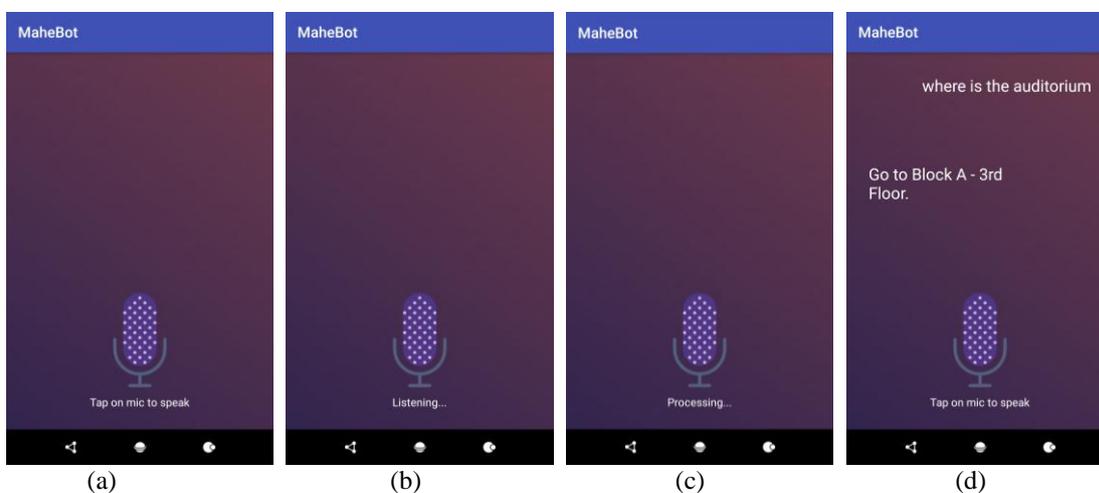


Figure 2 (a) Home Screen (b) Screen when the user is speaking (c) Screen when the query is being processed (d) Screen with the query and answer

RESULTS

The application home screen is as shown in Figure 2. The user can tap on the mic icon to begin speaking. Once the user finishes speaking, the bot processes the query and returns the result within 3 seconds. The answer is then both displayed and spoken out by the bot.

CONCLUSIONS AND FUTURE SCOPE

The Virtual Assistant, while useful for information and easy to use can also be improved in a few ways to make it more usable and accessible -

- *Expanding the scope:* The app is currently developed to find information about our institute. It can potentially be used by other universities, areas or offices as well.
- *Registering foreign languages and names:* Voice Assistant is currently only usable in English and cannot record words in another language including names outside of its vocabulary.
- *New functionalities:* In the future, this app can incorporate several new features as it progresses such as -
 - Making appointments
 - Registering for events
 - Making calls and sending emails

REFERENCES

- [1] “Fundamentals of Software Engineering”, Fourth Edition, Rajib Mall
- [2] <https://developer.apple.com/documentation/sirikit>
- [3] <https://www.dashbot.io/docs/google/outgoing-intents/>
- [4] <https://dialogflow.com/docs/agents> :
<https://dialogflow.com/docs>
- [5] <https://youtu.be/tdqkRwOb1s8>
- [6] <https://youtu.be/Nmrp7BbzkGc>